Session 119

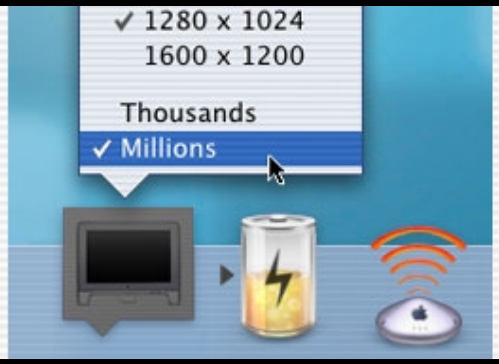
Carbon Controls and Appearance I

Guy Fullerton
High Level Toolbox

Overview

- Aqua adoption techniques
- Avoiding pitfalls
- Solving interface glitches

Demo



White Halos

- Side effect of Aqua shadows
- Controls erase their background
- Background color/pattern needs to be set up

Exploit Control Embedding

- Background based on parent controls
- System controls set up the background:
  - Tabs, Group Box, Placard, User Pane
- Root Control uses the Window's background:
  - SetThemeWindowBackground()

Exploit Control Embedding

- Custom CDEFs can set up the background too
- kEventControlApplyBackground
- kControlHasSpecialBackground and kControlMsgSetUpBackground

SetControlColorProc()

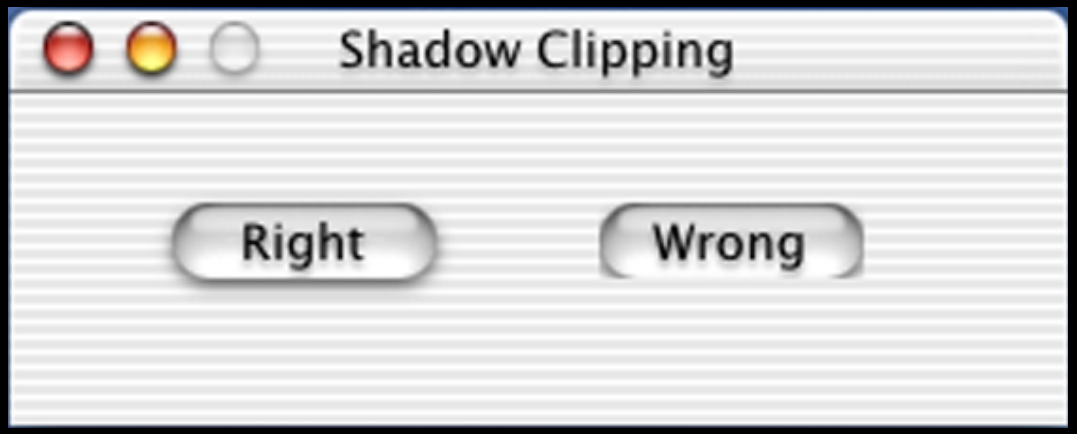
- Associates a ControlColorProc() with a ControlRef
- Overrides the embedding hierarchy's color choice
- Also allows text color customization

Prepare the Port

- When all else fails, Control Manager uses the background from the control's window port
- Set the port's color and pattern before calling the Control Manager
  - SetThemeBackground()
- Less than ideal for animating controls

Shadow Clipping

- Caused by clipping to the control's bounds
- In Aqua, controls draw outside their bounds
- PowerPlant is the obvious victim



Clip to the Structure Region

- Don't clip to the control's bounds
- Structure Region is the total draw area
  - Includes shadow
  - GetControlRegion() & kControlStructureMetaPart
- Many other regions supported

Focus Ring Issues

- Too dark
- Wreck widget structure
- Focus rings are drawn with alpha
- Need to “erase” beforehand

Control Manager Origin Issues

- Wrong animation location
- Misaligned background patterns
- Control Manager needs a 0,0 origin

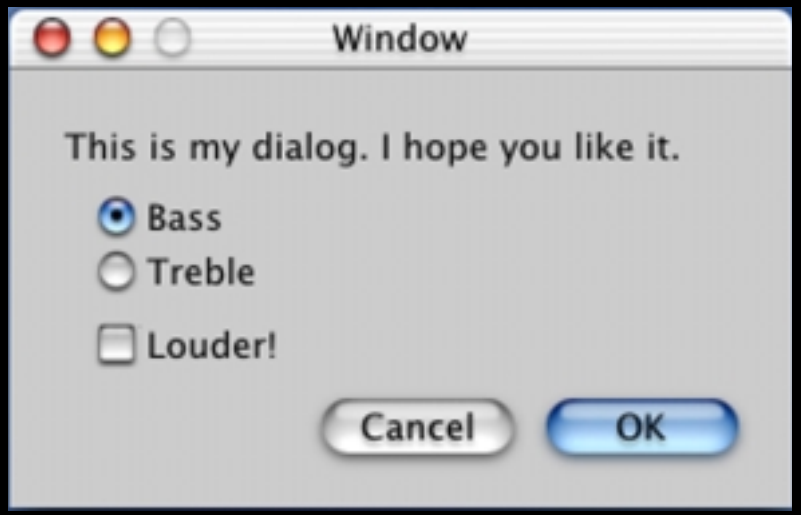
Control Manager Requirement

- Non 0,0 origin worked on Mac OS 9, right?
- Nearly impossible to leave the origin changed on Mac OS X:
  - Animating controls
  - Aqua backgrounds are patterns

Safe Origin Changes

- Ensure control bounds are always 0,0 relative
- Save and Restore around your origin changes:
  - GetPortBounds()
  - SetOrigin()

Pattern Issues



- Wanted a pattern, got a color!
- Wanted a color, got a pattern!
- Undesired gray or white backgrounds

Patterns vs. Colors

- Backgrounds may be patterns or colors
- Avoid unexpected patterns while drawing:
  - NormalizeThemeDrawingState()
- Need to save and restore both colors and patterns:
  - SaveThemeDrawingState() & RestoreThemeDrawingState()

Drawing Patterns

- Many ThemeBrush and ThemeBackgrounds available in Appearance.h
- SetThemeBackground() & EraseRect()
- SetThemePen() & PaintRect()
- SetThemeWindowBackground()
- ApplyThemeBackground()

ApplyThemeBackground()

- Similar to SetThemeBackground()
- Used for widgets which draw a background relative to a content rectangle:
  - Tabs, Group Box
- Aligns pattern to the content rectangle

Non-anti-aliased Text

- Unattractive in Aqua
- Unicode unfriendly
- Caused by using old QuickDraw text rendering routines:
  - DrawText(), DrawString(), TETextBox(), GetFontInfo()

Drawing Anti-aliased Text

- Use system CDEFs, MDEFs, WDEFs
  - StaticText control
  - UnicodeEditText control
- DrawThemeTextBox() and friends
- ATSUI and MLTE

ThemeText APIs

- Rendering: DrawThemeTextBox()
- Measuring: GetThemeTextDimensions(), GetThemeTextShadowOutset()
- Truncation: TruncateThemeText()
- Easy replacement for DrawText(), TextWidth(), TETextBox(), etc.

ThemeFontIDs

- Standard Metafonts:
  - System font, Views font, Push Button font, Window Title font, etc.
- Imply font, size, style
- Shadowing effects
- Unicode savvy

Customizing the Font and Style

- kThemeCurrentPortFont
- Supports QuickDraw font IDs
- Supports all styles except Outline and Shadow
- No Aqua shadows
- Inferior Unicode support

Theme Text Colors

- ThemeFontIDs do not imply a color
- Use SetThemeTextColor() to get the correct Aqua color before drawing

ThemeText Performance

- Use the CGContextRef parameter
- Avoid redundant/unnecessary calls
- Cache output values
- We're making things faster…

Appearance Manager

- Can draw most Aqua widgets via primitives
- Draws with alpha
  - Draw the background before calling APIs, or
  - Use ThemeEraseProcs
- GetThemeButtonBackgroundBounds()
- GetThemeMetrics()

Which Manager to Use?

- Control Manager is the best choice
- Appearance Manager works pretty well
- Real Controls work better:
  - Default Button Animation
  - Scroll Bar “Click to Here”

What Now?

- Start Aquafying!
- Pitfalls are surmountable
- Carbon High Level Toolbox is a great way to deliver Aqua applications

Roadmap

120 Controls & Appearance II: Stick around for more!	Hall 2 **Thurs., 10:30am**
FF114 Feedback Forum: High-Level Toolbox: Give us your thoughts	Room C **Fri., 3:30pm**
106 Carbon Event Manager: Everything you need to know about Carbon Events	Hall 2 **Tues., 5:00pm**
112 Carbon: Windows & Menus: Sheets, Window Groups, Menu Enhancements	Hall 2 **Wed., 2:00pm**

Roadmap

130 Designing & Using Aqua Icons	Hall 2 **Fri., 10:30am**
125 Apple Help	Room J2 **Thurs., 2:00pm**
127 Speech Recognition & Synthesis In-Depth	Room C **Thurs., 10:30pm**
114 Application Packaging & Document Binding	Hall 2 **Wed., 5:00pm**

User Experience Roadmap

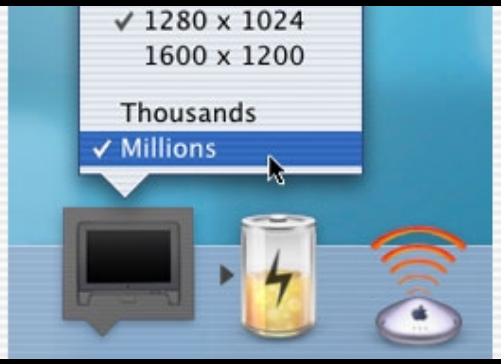
702 Interface Builder	Hall 2 **Tues., 3:30pm**
FF103 Feedback Forum: Aqua	Room J1 **Wed., 3:30pm**
FF114 Feedback Forum: High Level Toolbox	Room C **Fri., 3:30pm**

Who to Contact

John Geleynse
User Experience Technology Manager
Apple Worldwide Developer Relations
geleynse@apple.com

http://developer.apple.com/events/urls.html

Q&A



John Geleynse
User Experience Technology Manager
geleynse@apple.com

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001